IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A computer-implemented method for creating a graphical program, the method comprising:

creating a first graphical program using a graphical programming development environment, wherein said creating comprises interconnecting at least two of a first plurality of graphical program nodes or icons, wherein the first graphical program comprises the first plurality of interconnected graphical program nodes or icons which graphically represents functionality of the first graphical program, and wherein the first graphical program is executable by a computer system to perform the functionality;

storing the first graphical program in a memory; and

associating a debugging graphical program at a debugging location in the first graphical program, wherein said associating does not modify the functionality of the first graphical program, wherein the debugging graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the debugging graphical program~~, and wherein the debugging graphical program is executable by the computer system to perform the functionality~~ wherein the debugging graphical program was created using the graphical programming development environment;

wherein the debugging graphical program is executable during execution of the first graphical program to aid in debugging at least a portion of the first graphical program.

2. (Previously Presented) The computer-implemented method of claim 1, wherein said associating does not require a re-compilation of the first graphical program.

3. (Previously Presented) The computer-implemented method of claim 1, further comprising:

executing the first graphical program up to the debugging location;

executing the debugging graphical program after executing the first graphical program up to the debugging location; and

the debugging graphical program generating debugging results, wherein the debugging results are useful in analyzing at least a portion of the first graphical program.

4. (Previously Presented) The computer-implemented method of claim 3, further comprising:

completing execution of the first graphical program based on the debugging results of said executing the debugging graphical program.

5. (Previously Presented) The computer-implemented method of claim 3,

wherein said executing the debugging graphical program includes displaying the debugging results of the debugging graphical program.

6. (Previously Presented) The computer-implemented method of claim 3, wherein said executing the debugging graphical program comprises:

receiving data from the first graphical program; and

performing one or more of:

displaying the data from the first graphical program; and/or

logging the data from the first graphical program to a file.

7. (Previously Presented) The computer-implemented method of claim 3, wherein said executing the debugging graphical program comprises:

receiving data from the first graphical program;

generating statistics based on the received data; and

displaying the statistics.

8. (Previously Presented) The computer-implemented method of claim 7, wherein said statistics comprise one or more of:

data generated by the debugging graphical program;

data generated by a plurality of executions of the debugging graphical program during a corresponding plurality of executions of the first graphical program, wherein said data generated by the plurality of executions of the debugging graphical program includes differences in execution times between the plurality of executions of the debugging graphical program, wherein said differences in execution times are useable in optimizing performance the first graphical program.

9. (Previously Presented) The computer-implemented method of claim 3,

wherein said completing execution of the first graphical program is performed in a single stepping mode based on the debugging results of said executing the debugging graphical program.

10. (Previously Presented) The computer-implemented method of claim 1, further comprising:

executing the first graphical program up to the debugging location, wherein the first graphical program generates data at the debugging location;

providing the data to the debugging graphical program;

executing the debugging graphical program, wherein the debugging graphical program uses the data;

the debugging graphical program generating debugging results;

based on the debugging results, performing one or more of: halting execution of the first graphical program; entering single stepping mode in the first graphical program; and/or completing execution of the first graphical program.

11. (Previously Presented) The computer-implemented method of claim 10, wherein the first graphical program executes up to the debugging location where the debugging graphical program is associated, and waits for user input.

12. (Cancelled)

13. (Previously Presented) The computer-implemented method of claim 1,

wherein the first graphical program comprises a plurality of data flow paths;

wherein said associating the debugging graphical program at the location in the first graphical program comprises associating the debugging graphical program at a first data flow path in the first graphical program.

14. (Previously Presented) The computer-implemented method of claim 13, wherein said associating comprises:

storing information in at least one data structure, wherein the information comprises information regarding the first graphical program, the debugging graphical program, and the location where the debugging graphical program is attached along the first data flow path of the first graphical program.

15. (Previously Presented) The computer-implemented method of claim 13, wherein said associating comprises:

receiving user input from a pointing device selecting the first data flow path in the first graphical program, wherein the first data flow path is configured to carry data of a first data type;

displaying a plurality of debugging graphical programs, wherein each of the plurality of debugging graphical programs is compatible with the first data type, and wherein the plurality debugging graphical programs comprises the debugging graphical program; and

receiving user input selecting the debugging graphical program from the plurality of debugging graphical programs.

16. (Previously Presented) The computer-implemented method of claim 13, wherein said associating comprises:

receiving user input selecting the first data flow path in the first graphical program, wherein the first data flow path is configured to carry data of a first data type;

determining the first data type of the first data flow path;

displaying a plurality of debugging graphical programs appropriate for the first data type of the first data flow path; and

receiving user input selecting the debugging graphical program from the plurality of debugging graphical programs appropriate for the first data type of the first data flow path.

17. (Previously Presented) The computer-implemented method of claim 1, wherein said associating the debugging graphical program at the debugging location in the first graphical program comprises associating the debugging graphical program at a node or icon in the first graphical program.

18. (Previously Presented) The computer-implemented method of claim 1, further comprising:

disassociating the debugging graphical program from the first graphical program, wherein said disassociating does not modify the first graphical program and/or does not require a re-compilation of the first graphical program.

19. (Previously Presented) The computer-implemented method of claim 1,

wherein the first graphical program is located on a first computer system;

wherein the debugging graphical program is located on a second computer system, wherein the second computer system is coupled to the first computer system over a network.

20. (Previously Presented) The computer-implemented method of claim 19, the method further comprising:

executing the first graphical program on the first computer system up to the debugging location;

executing the debugging graphical program on the second computer system, wherein the debugging graphical program is executed after executing the first graphical program on the first computer system up to the debugging location;

the debugging graphical program generating debugging results on the second computer system; and

providing the debugging results from the second computer system to the first computer system.

21. (Previously Presented) The computer-implemented method of claim 1,

wherein the first graphical program is located on a first computer system, wherein the first computer system is a target computer system coupled to or comprised in a second computer system;

wherein the debugging graphical program is located on and executed on the first computer system.

22. (Previously Presented) The computer-implemented method of claim 1,

wherein the first graphical program is located on a first computer system, wherein the first computer system is a target computer system coupled to or comprised in a second computer system;

wherein the debugging graphical program is located on and executed on the second computer system.

23. (Currently Amended) A computer-implemented method for executing a first graphical program, the method comprising:

executing the first graphical program up to a debugging location, wherein the first graphical program comprises a first plurality of interconnected graphical program nodes or icons that graphically represents functionality of the first graphical program, and wherein the first graphical program is executable by a computer system to perform the functionality, wherein the first graphical program generates data at the debugging location, wherein the first graphical program was created using a graphical programming development environment;

providing the data to a debugging graphical program, wherein the debugging graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the debugging graphical program~~, and wherein the debugging graphical program is executable by the computer~~

~~system to perform the functionality~~ wherein the debugging graphical program was created using the graphical programming development environment;

executing the debugging graphical program, wherein the debugging graphical program uses the data;

the debugging graphical program generating debugging results;

wherein use of the debugging graphical program does not require modification or re-compilation of the first graphical program.

24. (Previously Presented) The computer-implemented method of claim 23,

after the debugging graphical program generates debugging results, performing one or more of: halting execution of the first graphical program; entering a single stepping mode in the first graphical program; and/or completing execution of the first graphical program.

25. (Previously Presented) The computer-implemented method of claim 23, further comprising:

associating the debugging graphical program at the debugging location in the first graphical program;

wherein said associating does not require modification or recompilation of the first graphical program.

26. (Previously Presented) The computer-implemented method of claim 23,

wherein the first graphical program is located on a first computer system;

wherein the debugging graphical program is located on a second computer system, wherein the second computer system is coupled to the first computer system over a network;

wherein the first graphical program executes on the first computer system up to the debugging location;

wherein the debugging graphical program executes on the second computer system, wherein the debugging graphical program is executed after executing the first graphical program on the first computer system up to the debugging location; and

wherein the debugging graphical program generates debugging results on the second computer system.

27. (Currently Amended) A computer-implemented method for analyzing a first graphical program, the method comprising:

storing the first graphical program in a memory of a computer system, wherein the first graphical program comprises a first plurality of interconnected graphical program nodes or icons that graphically represents functionality of the first graphical program, and wherein the first graphical program is executable by the computer system to perform the functionality, wherein the first graphical program was created using a graphical programming development environment;

associating a second graphical program at a location in the first graphical program, wherein said associating does not modify the functionality of the first graphical program, wherein the second graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the second graphical program~~, and wherein the second graphical program is executable by the computer system to perform the functionality~~, wherein the second graphical program was created using the graphical programming development environment;

wherein the second graphical program is executable during execution of the first graphical program to aid in analyzing at least a portion of the first graphical program.

28. (Previously Presented) The computer-implemented method of claim 27, wherein said associating does not require a re-compilation of the first graphical program.

29. (Previously Presented) The computer-implemented method of claim 27, further comprising:

executing the first graphical program up to the location;

executing the second graphical program after executing the first graphical program up to the location; and

the second graphical program generating results, wherein the results are useful in analyzing at least a portion of the first graphical program.

30. (Previously Presented) The computer-implemented method of claim 27, further comprising:

executing the first graphical program up to the location, wherein the first graphical program generates data at the location;

providing the data to the second graphical program;

executing the second graphical program, wherein the second graphical program uses the data;

the second graphical program generating results;

based on the results, performing one or more of: halting execution of the first graphical program; entering a single stepping mode in the first graphical program; and/or completing execution of the first graphical program.

31. (Currently Amended) A memory medium comprising program instructions for analyzing a first graphical program, wherein the program instructions are executable to implement:

storing the first graphical program in a memory of a computer system, wherein the first graphical program comprises a first plurality of interconnected graphical program nodes or icons that graphically represents functionality of the first graphical program, and wherein the first graphical program is executable by the computer system to perform the functionality, wherein the first graphical program was created using a graphical programming development environment;

associating a second graphical program at a location in the first graphical program, wherein said associating does not modify the functionality of the first graphical program, wherein the second graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the second graphical program~~, and wherein the second graphical program is executable by the computer system to perform the functionality~~, wherein the second graphical program was created using the graphical programming development environment;

wherein the second graphical program is executable during execution of the first graphical program to aid in analyzing at least a portion of the first graphical program.

32. (Original) The memory medium of claim 31, wherein said associating does not require a re-compilation of the first graphical program.

33. (Original) The memory medium of claim 31, wherein the program instructions are further executable to implement:

executing the first graphical program up to the location;

executing the second graphical program after executing the first graphical program up to the location; and

the second graphical program generating results, wherein the results are useful in analyzing at least a portion of the first graphical program.

34. (Previously Presented) The memory medium of claim 31, wherein the program instructions are further executable to implement:

executing the first graphical program up to the location, wherein the first graphical program generates data at the location;

providing the data to the second graphical program;

executing the second graphical program, wherein the second graphical program uses the data;

the second graphical program generating results;

based on the results, performing one or more of: halting execution of the first graphical program; entering a single stepping mode in the first graphical program; and/or completing execution of the first graphical program.

35. (Currently Amended) A memory medium comprising:

a first graphical program, wherein the first graphical program comprises a first plurality of interconnected nodes which visually indicate functionality of the first graphical program, and wherein the first graphical program is executable by a computer system to perform the functionality, wherein the first graphical program was created using a graphical programming development environment;

a second graphical program, wherein the second graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the second graphical program~~, and wherein the second graphical program is executable by the computer system to perform the functionality~~, wherein the second graphical program was created using the graphical programming development environment;

a data structure which is operable to store information associating the second graphical program with a location in the first graphical program, wherein the functionality of the first graphical program is not modified by the second graphical program;

wherein the second graphical program is executable during execution of the first graphical program to aid in analyzing at least a portion of the first graphical program.

36. (Previously Presented) The memory medium of claim 35, further comprising program instructions which are executable to:

execute the first graphical program up to the location, wherein the first graphical program generates data at the location;

provide the data to the second graphical program;

execute the second graphical program, wherein the second graphical program uses the data and generates results;

based on the results, perform one or more of: halting execution of the first graphical program; entering single stepping mode in the first graphical program; and/or completing execution of the first graphical program.

37. (Previously Presented) The computer-implemented method of claim 1, wherein said associating the debugging graphical program at the debugging location in the first graphical program comprises:

including the debugging graphical program at the debugging location in the first graphical program.

38. (Previously Presented) The computer-implemented method of claim 23, further comprising:

including the debugging graphical program at a debugging location in the first graphical, wherein said including does not modify the functionality of the first graphical program.

39. (Previously Presented) The computer-implemented method of claim 27, wherein said associating the second graphical program at the location in the first graphical program comprises:

including the second graphical program at the location in the first graphical program.

40. (Previously Presented) The memory medium of claim 31, wherein, in said associating the second graphical program at the location in the first graphical program comprises, the program instructions are further executable to implement:

including the second graphical program at the location in the first graphical program.